NAS2-11530

# RIACS Summer Review 1985

*Peter J. Denning*

October 1985

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS TR 85.13



RIACS

Research Institute for Advanced Computer Science

# RIACS Summer Review 1985

*Peter J. Denning*

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS TR 85.13
October 1985

*Attached is a short summary of RIACS and its accomplishments for 1985 plus the slides used at the oral review for Ames Center management on October 11, 1985.*

---

**Work reported herein was supported in part by Contract NAS2-11530 from the National Aeronautics and Space Administration (NASA) to the Universities Space Research Association (USRA).**

# RIACS Summer Review 1985

Peter J. Denning
Research Insitute for Advanced Computer Science

October 1985

RIACS was established at the Ames Research Center in June 1983. It is privately operated by the Universities Space Research Association (USRA) under contract to NASA. Its purposes are to help build a pool of top computer science talent at Ames, to conduct a research program, to attract top computer science faculty and industrial researchers on leave, and to establish links with universities and industry.

As of September 1985, RIACS had a core staff of 12 persons including 8 scientists, and a supplementary staff of 18 persons, including 5 summer students. The supplementary staff were paid from additional funds provided by Task Orders.

RIACS also operates an advanced computing facility consisting of a Sequent Balance 8000, 12-processor UNIX machine, a Silicon Graphics IRIS workstation, a Ridge 32 computer (used as document server), a QMS laser printer, and Intel iPSC hypercube, a VAX 11/730 (used as a network gateway to ARPANET, CSNET, and UUCP), an image reader, and a MacIntosh with hard disk and direct connection to the Sequent. All these components are attached to an Ethernet to which new systems can be connected in short order.

The RIACS research program focuses on projects in concurrent processing and autonomous systems, two areas of computer science of top priority to NASA. The goal is advanced software systems that support scientific research in many disciplines, from problem formulation to results dissemination. The concurrent processing research is centered on a testbed of multiprocessors, initially the Sequent and the Intel hypercube, coupled with a program to form interdisciplinary teams to study algorithms in disciplines where computing power in excess of 1 GFLOPS is required. Initial software tools for this effort include Concurrent C and a Graphical Shell. Concurrent C is an extension of the C

language allowing programming of parallel tasks. The Graphical Shell is an extension of the UNIX shell that permits computations consisting of distributed processes to be expressed as dataflow graphs drawn by the user. Because the processes can be on different machines, the system will easily support workstation-supercomputer computations. The autonomous systems research is centered on a model of sparse, distributed memory capable of accessing stored patterns when presented with partial patterns. The patterns can be very large, 1000 bits in the prototype. By 1986 we will have a simulator of the memory operational and in use for robotics and human factors research. In cooperation with Stanford, we will develop breadboard models of the memory device.

# RIACS

Program Review

Prepared August 1985

Presented October 11, 1985

# PURPOSES

- Help build pool of top CS talent at Ames
- Core research program
- Attract top CS faculty on sabbaticals
- Attract industry researchers on sabbaticals
- Links to universities and industry
- Transfer technology

RIACS started operation June 1983.

# FUNDING MECHANISM

(NASA → USRA)

**Core ~$1M in FY85 and FY86**

**Task ~$800K in FY85**

Changing to Cooperative Agreement through
University Affairs Office
upon renewal in January 1986.

# ACCOMPLISHMENTS

- Computer Laboratory
- Software Tools
- Staff Buildup
- Tasks
- Technical Reports
- Summer Students
- Outside Contacts
- Research Program (Project "R")

# Computer Laboratory

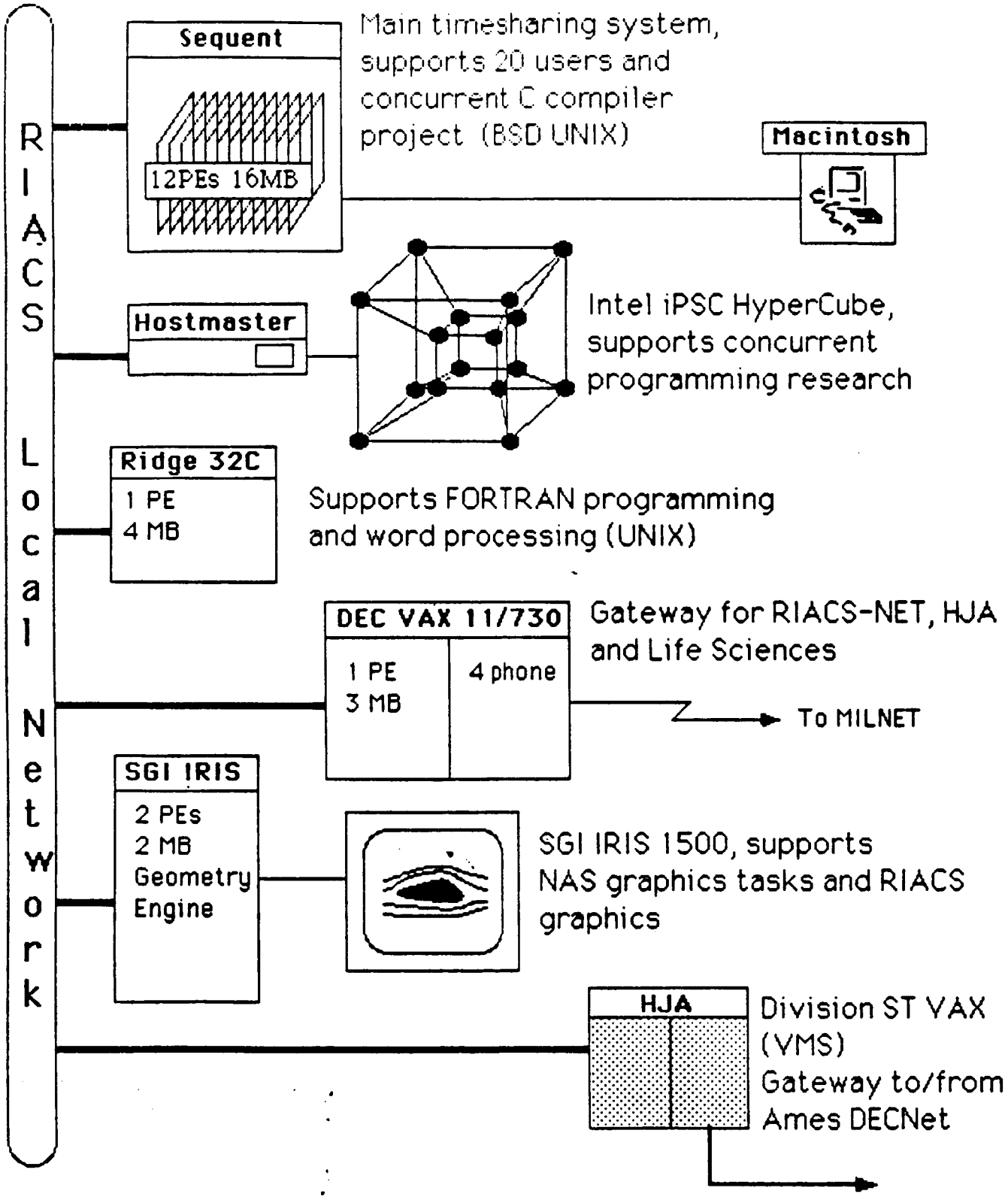
- Sequent Balance 8000 12–processor system
- Silicon Graphics IRIS [NAS Task]
- Ridge 32 Computer (Document server)
- QMS Laser Printer
- Intel iPSC hypercube
- VAX 11/730 network gateway (ARPANET, CSNET, uucp)
- Image reader
- MacIntosh with hard disk and QMS Laserwriter simulator

- UNIX BSD operating system (+ System 5)

The diagram on the next page shows the components of the RIACS computing facility. The Sequent Balance 8000 computer has 12 processors, 16 million bytes of main memory, and runs Berkeley 4.2 UNIX. Twenty active users are connected to this system; most have BitGraph terminals. The MacIntosh system's printer port is connected to a Sequent port linked to process that generates Imagewriter output on the QMS laserprinter. The 32-node Intel iPSC hypercube is connected to a small computer, the 310 Hostmaster, running a UNIX variant called Xenix. The Ridge 32 computer is a document server that runs "troff" jobs and drives the QMS laserprinter. The DEC VAX-11/730 is used as a communications gateway, linking the facility to MILNET, phone lines, and the UUCP net. The Silicon Graphics IRIS workstation is part of a project to port UNIX utilities into the workstations used by the Numerical Aerodynamic Simulator (NAS) facility at Ames. "HJA" are the initials of a VAX-11/780 in the Thermosciences Division of Ames. By linking these dissimilar units on to a common Ethernet, the RIACS laboratory has demonstrated its ability to set up and maintain a heterogeneous computing environment.

# RIACS Computing Facility

## Hetergeneous Network



# Library of Software Tools

- Address database + office automation
- Windows, pop–up menus, mouse for EMACS on Bitgraphs
- Laser printer fonts
- Concurrent C
- Graphics Shell
- Network interfaces for MacIntosh; output can be directed to Laser Printer or troff.

The following two pages are images retrieved from the screens of the Bitgraph terminals, illustrating some home-grown software for office automation. The first picture shows two windows in the emacs editor. The lower window contains a letter to John Bruno; to overcome the complexity of dealing with the UNIX 'troff' program, our system presents a standard letter template whose address is filled in automaticaly from a database and whose body is typed by the user. The second image is the letter printed on the laser printer about 30 seconds after Bob Brown gave the print command.

```
chong     ames-hjulian:txd5 Oct  7 13:52 :26
gl        hydra:ttyi2       Oct  7 08:10
komornic  ames-hjulian:txa2 Oct  7 14:57 :32
lars      ames-hjulian:txc1 Oct  7 12:49 :03
peda      ames-hjulian:txb0 Oct  7 19:37 :17
rlb       hydra:ttyh5       Oct  4 16:49
rlb       hydra:ttyh6       Oct  7 17:49
root      daedalus:disp0    Oct  7 15:12
tannehil  ames-hjulian:txe5 Oct  7 19:55
(hydra)
```

```
csh* (Bottom) 7:05pm 0.11 [7] <41.1%-> (Normal)
..n -t 2
.ll 6i
.ie n .po 0.5i
.el .po 1.25i
.na
.nh
.ps 12
.vs 14
.de pp
.sp 1
.ti 5n
..
.de us \" underlined string
\\$1\l'|0\(ul'
..
.wh -1i pb
.de pb
'bp
'sp 1i
..
.sp 2.i
.ti 4i
October 4, 1985
.sp 2
.nf
Prof. John Bruno
Department of Computer Science
Engineering 2106
University of California
Santa Barbara, CA  93106
.fi
.sp 2
Dear John,
.pp
The Hypercube has arrived and is up and running.
I have written a few programs for it and between myself and Nancy we
have developed some Makefiles that simplify the program development
process.
.pp
Give Joan a call to get yourself an account set up.
She'll also provide access information that will allow you to work on
your task from your office at UCSB.
.sp 1
.in 4i
Cordially,
.sp 4
Robert L. Brown
.br
Scientist
.in 0
,Bruno.1 (97%) 7:05pm 0.11 [7] <41.1%-> (Text)
```

```
TEXT
troff
dplot
print
select
troff(
dplot(
PRINT..
WORD..
```

```
WORD PROCESSING
make-letter
make-temp-file
insert-letter
insert-memo
labels
type-labels
network
form letter
rebuild form letter
MENU...
```

Mail Stop 230-5
NASA Ames Research Center
Moffett Field, CA 94035
(415) 694-6363

October 4, 1985

Prof. John Bruno
Department of Computer Science
Engineering 2106
University of California
Santa Barbara, CA 93106

Dear John,

The Hypercube has arrived and is up and running. I have written a few programs for it and between myself and Nancy we have developed some Makefiles that simplify the program development process.

Give Joan a call to get yourself an account set up. She'll also provide access information that will allow you to work on your task from your office at UCSB.

Cordially,

Robert L. Brown
Scientist

# STAFF

## (End FY85)

| | |
|---|---|
| PLAN | 11 core<br>20 others |
| ACTUAL | 12 core<br>18 others |

Core -- Paid from core funds
(Excludes summer students)
7 PhD
1 BS
4 Support

**Core fully committed**

# STAFF

| | | | |
|---|---|---|---|
| SENIOR SCIENTIST | Director | Denning | |
| | Chief Sci | Raugh | |
| | Sr Sci | Leiner | |
| SCIENTIST | Engineer | Adams | |
| | Scientist | Brown | |
| | Scientist | Bishop | *T* |
| | MTS | Briggs | |
| | Scientist | Cheeseman | *T* |
| | Scientist | Johnson | *T* |
| | Scientist | Kanerva | *T* |
| | Scientist | Levin | *T* |
| | Scientist | Partridge | *T* |
| VISITOR | Sabbatical | Patrick | |
| | Sabbatical | Sevcik | |
| SUPPORT | Administrator | Ciraulo | |
| | Facilities Coord | Elliott | |
| | Syst Progr | Blachman | |
| | Secretary | Kohutanycz | |
| OTHER | Consultant | Bruno | *T* |
| | Consultant | Callahan | *T* |
| | Consultant | Flynn | *T* |
| | Consultant | Long | *T* |
| | MIT Consultant | Ackerman | *T* |
| | MIT Consultant | Dennis | *T* |
| | MIT Consultant | Gao | *T* |
| | Summer Stu | Barerra | |
| | Summer Stu | Bitar | |
| | Summer Stu | Doyle | |
| | Summer Stu | Ruff | |
| | Summer Stu | Yost | |

# EXISTING TASKS

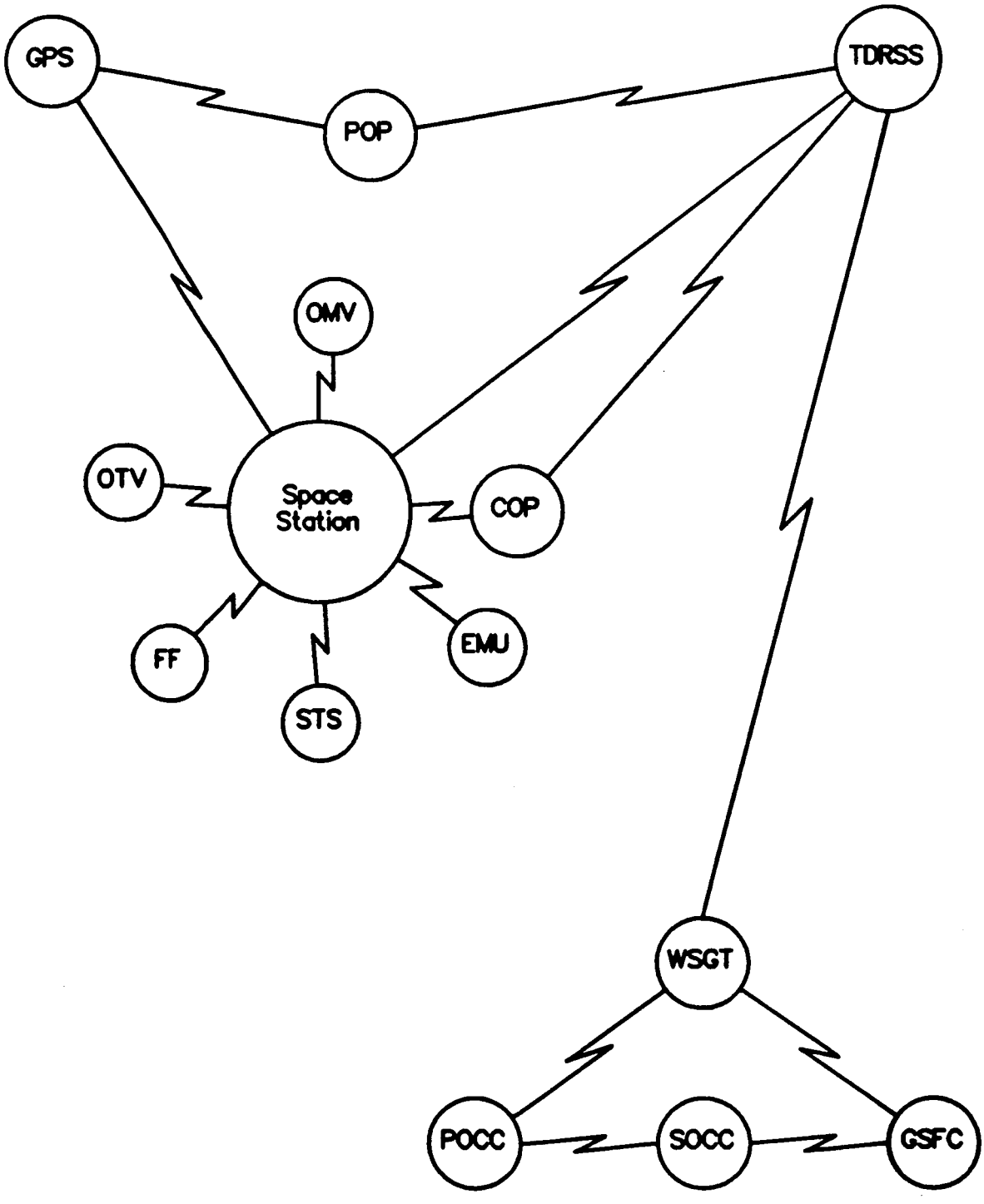
## (Start 1984 and end in 1985)

| | | |
|---|---|---|
| NAS Technical Studies | **Levin** | Bailey (NAS) |
| Comp'l Chemistry | **Levin** | Arnold (Adv Comp'l Concepts) |
| NAS Perf Modeling | **Sevcik** | Hofman (NAS) |
| Graphics Workstation | **Bishop** | Lasinski (NAS) |
| Data Networks Concepts | **Johnson** | Grant (Space Science) |
| Data Flow Evaluation | **Adams Brown Denning** | Stevens (Comp Techn) |
| Tightly Coupled MP | **Callahan** | Arnold (Adv Comp'l Concepts) |
| Analytic Modeling | **Sevcik** | Grant (Space Science) |

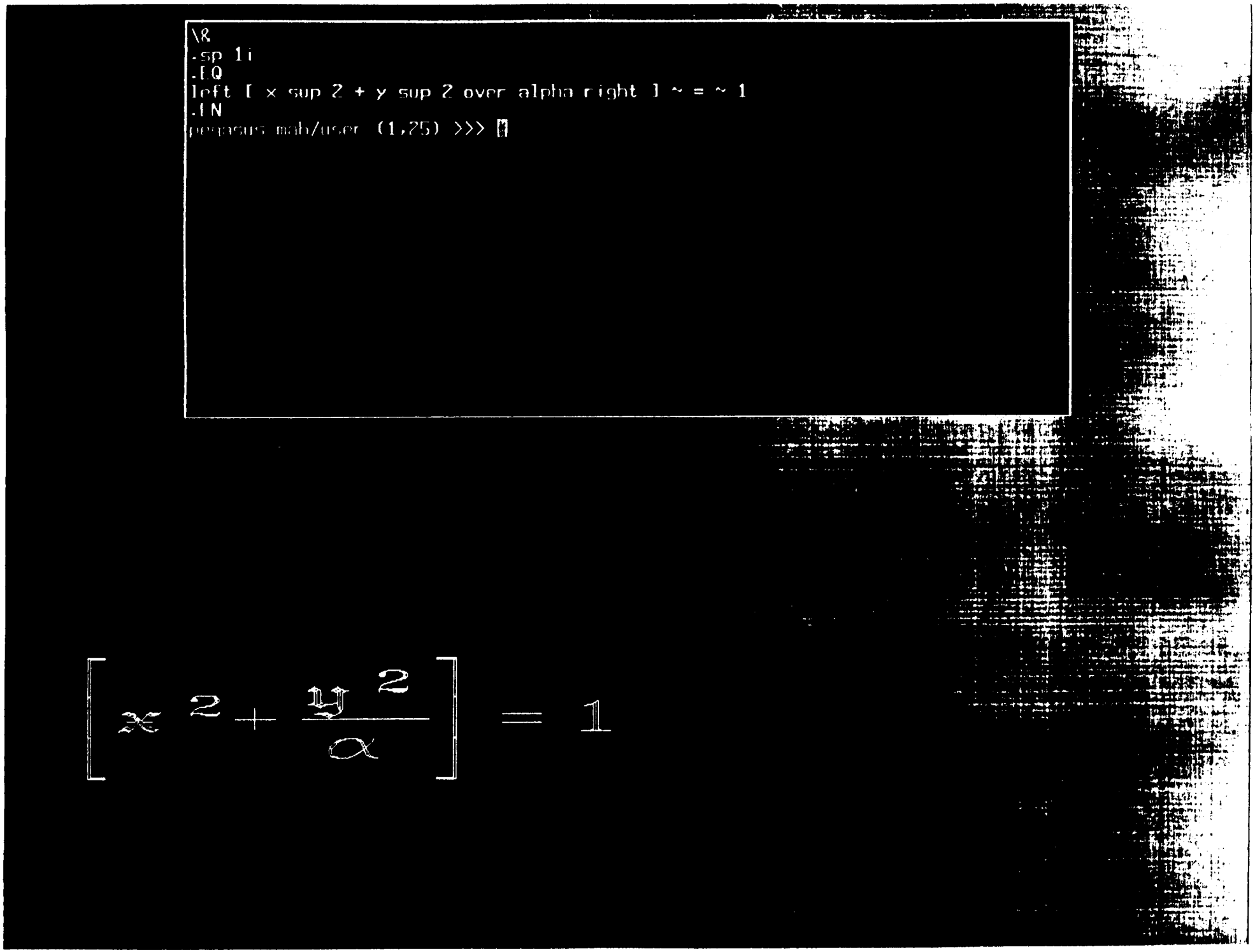
**Residual value = $211K**

The following two pages are images associated with two of the existing tasks. The first is a figure by Marjory Johnson on high-speed data networks for the space station. The diagram shows the space station surrounded by six platforms. This cluster is linked with three communication satellites (GPS, POP, TDRSS), which are in turn linked with a ground network.

The second figure is an image of an equation previewing system working on the Silicon Graphics IRIS workstation, part of a project by Matt Bishop. The inner window shows an equation in the language of the 'eqn' formatter used with 'troff.' This equation has been processed by eqn and troff to produce the output displayed in the outer window. The user can insert the equation text into the troff document after previewing.




Space Station System Communication Links at IOC


\R
.sp 1i
.EQ
left [ x sup 2 + y sup 2 over alpha right ] ~ = ~ 1
.EN
pegasus mab/user (1,25) >>> [ ]
$$\left[ x^2 + \frac{y^2}{\alpha} \right] = 1$$

# NEW TASKS

## (Start in 1985)

| | | |
|---|---|---|
| MOSAIC Workstations *(defunct)* | **Brown Johnson** | Grant (Space Science) |
| LES on Hypercube | **Bruno** | Arnold (Adv Comp'l Concepts) |
| RIACS/ISO Brochure *(defunct)* | **Adams Denning** | Lum (Space Science) |
| Analytic Modeling | **Sevcik** | Grant (Space Science) |
| Chem Algorithms | **Partridge** | Arnold (Adv Comp'l Concepts) |
| UNIX Conc Progr | **Brown** | Bailey (NAS) |
| Security Risk Anal | **Long** | Hofman (NAS) |
| Learning Research | **Cheeseman** | Lum (Space Science) |
| NAS Technical Studies | **Levin** | Bailey (NAS) |
| Chem Algorithms | **Levin** | Arnold (Adv Comp'l Concepts) |
| Hypercube | **Raugh** | Arnold (Adv Comp'l Concepts) |

**Face value = $753K**

The next page contains the article from IEEE *Computer* magazine, October 1985, announcing the installation of the first Intel iPSC hypercubes. The one at RIACS was sponsored through a new task begun in 1985.

# UPDATE

## Intel ships first commercial concurrent computers

The first customer shipments of its iPSC family of concurrent computers has been announced by Intel Corporation. Initial customers include Yale University, Oak Ridge National Laboratories, the Research Institute for Advanced Computer Science, and the Supercomputing Research Center.

According to Intel, the iPSC systems are the first commercially available computers to employ large-scale concurrent processing capabilities, although several other companies have announced the intention of building machines that employ concurrent processing.

The architecture of the iPSC family is based on the "hypercube" interconnect structure developed at the California Institute of Technology under funding by the US Department of Energy and the Defense Advanced Research Projects Agency. The hypercube architecture consists of multiple, independent computational nodes, each with memory, linked to other nodes by dedicated communications channels. Each computing node connects to as many as seven other nodes and includes one Intel 80286 central processing unit, an 80287 numeric processing unit, and 512K of memory. The point-to-point communications channels operate at 10 megabits per second.

The nodes' ability to work concurrently provides the processing power necessary for computationally intensive scientific applications, while the use of readily available VLSI components greatly lowers costs and helps researchers focus more closely on their specific applications, the company says.

Yale University has purchased the iPSC/d7, which has 128 computational nodes and 64 megabytes of distributed memory. Its Research Center for Scientific Computation will use the machine for research into such areas as multiprocessor operating systems, programming languages, algorithms, and applications. Application areas will include finite element computations, seismic processing, computational fluid mechanics, circuit analysis, and symbolic computing in such areas as artificial intelligence and VLSI design tools.

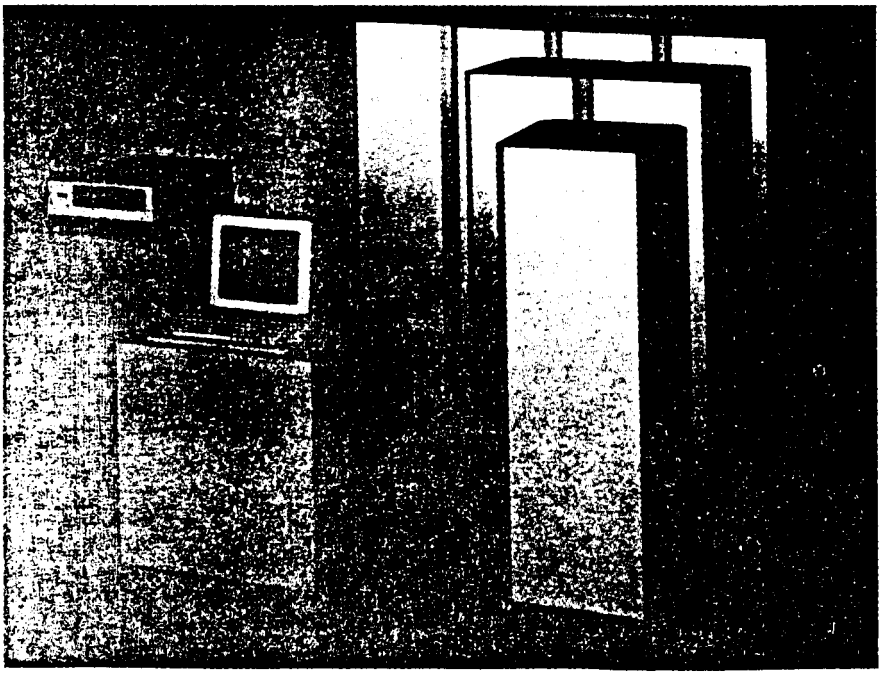
Oak Ridge National Laboratories will use the iPSC/d6, with 64 nodes and 32 megabytes of memory, in its Mathematical and Statistics Research Section to perform basic research, primarily for the US Department of Energy. Oak Ridge researchers will test parallel algorithms in computational and numerical analyses such as sparse matrices and partial differential equations. Specific applications will include chemistry, solid-state physics, environment, intelligence, specifically robotics.

The Research Institute for Advanced Computer Science, in cooperation with the NASA AMES Research Center, will use an iPSC/d5, with 32 nodes and 16 megabytes of memory, to develop parallel programming methodologies for problems in computational fluid dynamics and computational chemistry. NASA is sponsoring the research and will acquire the equipment.

The Supercomputing Research Center [illegible] advancing state-of-the-art supercomputing for national security purposes, including developing and evaluating parallel processing. The iPSC/d6 will be among its first scientific computers.

According to Intel, the iPSC family of computers are cost competitive with conventional supercomputers. The capability of conventional vector processing supercomputers is seldom achieved on typical scientific problems, Intel says. The Cray 1, for example, has a peak performance of more than 160 million floating-point operations per second, but users commonly report its performance to average from 10 MFLOPS to 35 MFLOPS, or an efficiency of about five to 20 percent. The Cray 1 costs in the range of $5 million to $10 million. The iPSC products, on the other hand, Intel says, have a performance range of from 2.5 MFLOPS to 10 MFLOPS, achieve an efficiency of 80 to 99 percent, and cost from $150,000 to $520,000.

The company feels that concurrent processing is the best long-term approach to achieving affordable, accessible supercomputing and that linking physically compact and widely available VLSI microprocessors is more practical than linking supercomputers, superminicomputers, or array processors.



**The Intel iPSC concurrent computer system includes the cube, the multiprocessor computational element that can be configured with 32, 64, or 128 nodes (free standing at right), and the cube manager, a Xenix-based Intel 286/310 microcomputer system (on boxes at left).**

# PENDING TASKS

## (Start in 1985)

| | | |
|---|---|---|
| Distributed Memory | **Kanerva** | Lum (Space Science) Nagel (Human Factors) |
| Network Security | **Bishop (60%)** | Hofman (NAS) |
| IRIS Workstation | **Bishop (40%)** | Lasinski (NAS) |
| Data Network Concepts | **Johnson** | Grant (Space Science) |
| Scientific Workstations | **Brown Johnson** | Grant (Space Science) |
| ..... | ..... | ..... |
| Advanced Graphics | **Gomez** | Hofman (NAS) |
| MP in Aircraft Env | TBD | Yee (Aircraft Systs) |
| Human Factors | **Blye** | Nagel (Human Factors) |

**Face value = $1.3M**

# WHO's BENEFITTED?

(Technology Transfer)

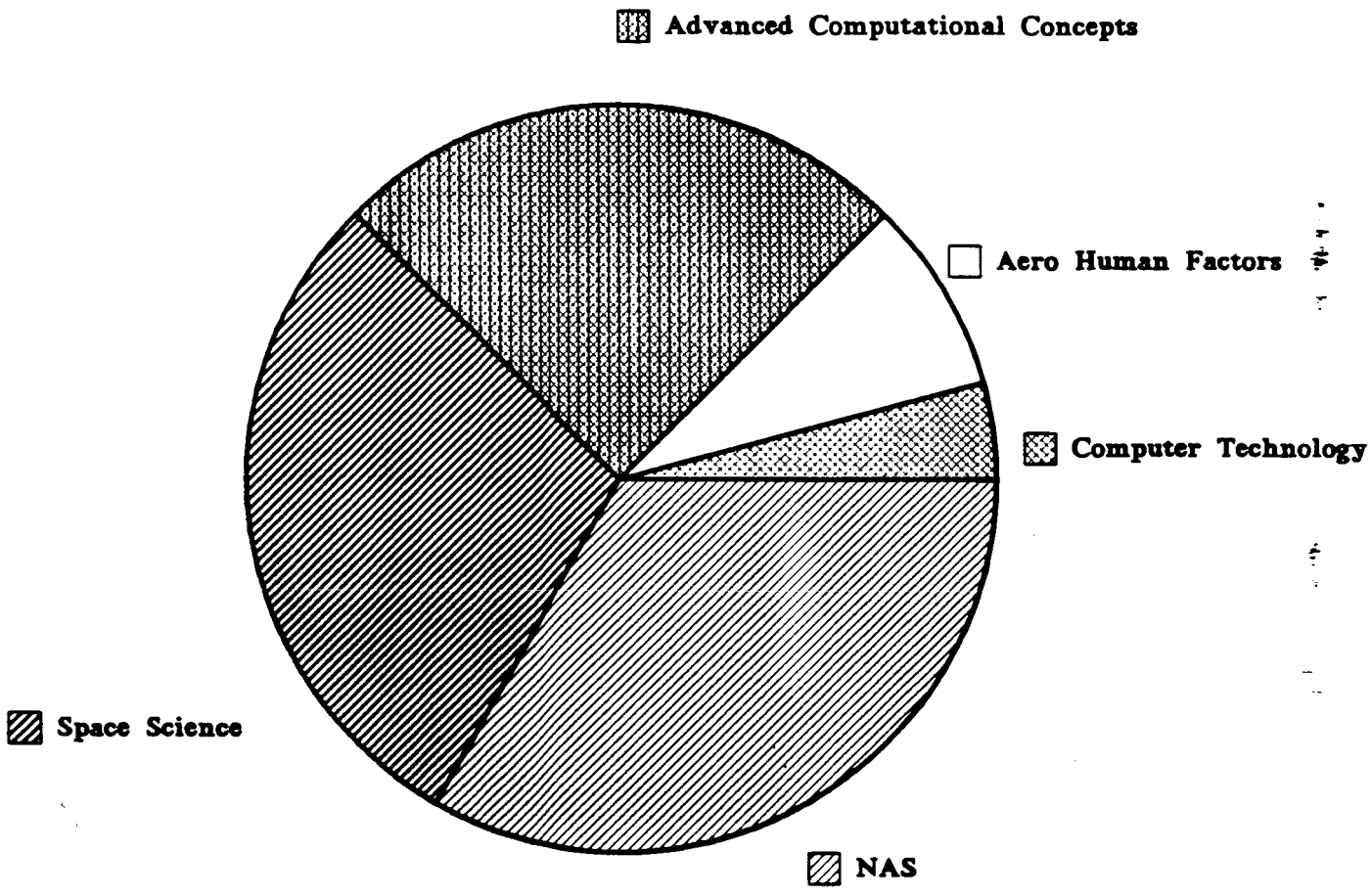
| | |
|---|---|
| NAS | 8 |
| Space Science | 7 |
| Adv Comp'l Concepts (CFD, CC) | 6 |
| Aero Human Factors | 2 |
| Computer Technology | 1 |
| TOTAL | 23 |

Sabbaticals create information flow:

Patrick Ames → Duke Univ

Sevcik Ames → U Toronto

# Who's Benefitted?



# TECHNICAL REPORTS

Totals:

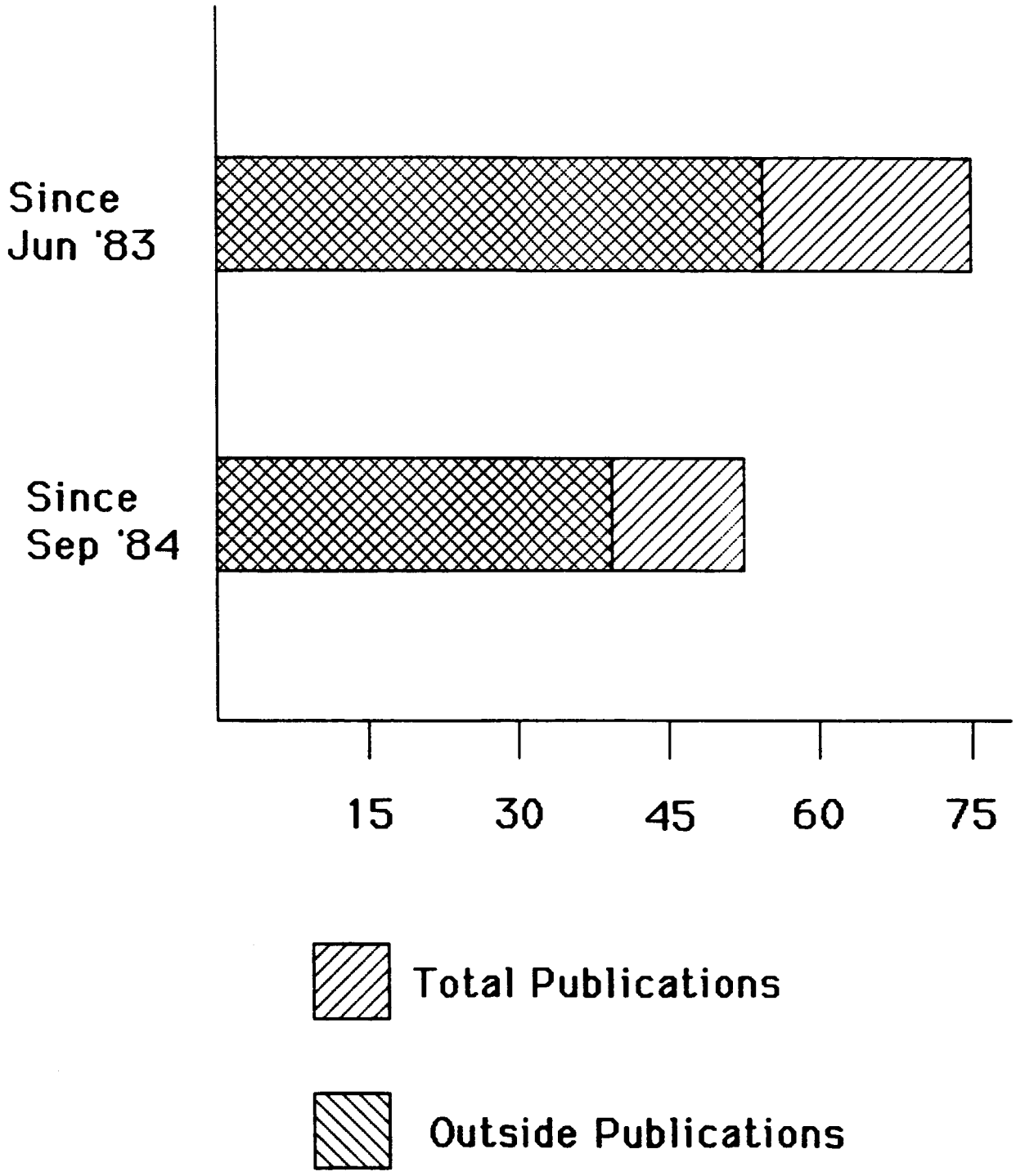
75 since June 1983
50 since September 1984

For Outside Publication:

55 since June 1983
39 since September 1984

12 journals, 10 conferences

# RIACS Publications



# TECHNICAL REPORTS (con't)

| NAME | # Papers | # Pages | # TRs | # Pages |
|---|---|---|---|---|
| Adams | 6 | 41 | 2 | 67 |
| Bishop | 0 | 0 | 2 | 49 |
| Briggs | 3 | 19 | 1 | 26 |
| Brown | 2 | 22 | 2 | 67 |
| Denning | 12 | 54 | 5 | 119 |
| Johnson | 2 | 23 | 2 | 46 |
| Levin | 1 | 3 | 1 | 6 |
| Partridge | 18 | 133 | 0 | 0 |
| Patrick | 1 | 10 | 1 | 44 |
| Sevcik | 1 | 5 | 2 | 36 |

# SUMMER STUDENTS

Joseph Barrera (Senior, Harvey Mudd College)
``Concurrent C translator & runtime support"
HYPERCUBE, MULTIPROCESSORS

Philip Bitar (Graduate student, UC Berkeley)
``Synchronization in MP systems"

Dee Doyle (Graduate student, NM State)
``Office automation software"
AVAILABLE TO ANY AMES UNIX INSTALLATION

Ritchey Ruff (Graduate student, U Oregon)
``Information theoretic curve fitting"

Jeff Yost (Graduate student, U Utah)
``Graphical shell"
WORKSTATION – SUPERCOMPUTER CONNECTIONS

The next page is a Concurrent C program written by Joseph Barrera as part of his summer project. Under supervision by Bob Brown, Barrera developed a preprocessor that converts the new syntax (e.g., **process spec**, **accept**, **create**) into procedure calls within standard C. The procedure calls invoke operations in a modified UNIX kernel that implement fast process creation and semaphores. This program runs on the Sequent Balance 8000 at RIACS. An extension of the runtime support system is being developed by Nancy Blachman. It will run on the Intel hypercube.

```
#include <stdio.h>
#define BETTING_LIMIT   1000

process spec bettor()
{
   trans void init(process bettor, int, int);
   trans void placebet(int);
};

process body bettor()
{
   int i_am_first, mylimit;
   int mynextbet = 1, oplastbet = 1;
   process bettor opponent;

   accept init(other_player, first_or_not, limit) {
       opponent = other_player;
       i_am_first = first_or_not;
       mylimit = limit;
   }
   if (i_am_first)
       opponent.placebet(mynextbet);
   while (mynextbet > 0 && oplastbet > 0) {
       accept placebet(bet) {
          oplastbet = bet;
       }
       if (oplastbet > 0) {
          mynextbet = oplastbet + 1 + rand()%100;
          if (oplastbet > 0)
              mynextbet=0;
          opponent.placebet(mynextbet);
       }
   }
   if (i_am_first) {
      if (mynextbet > 0)
          printf("I won: last bet %d\n", mynextbet);
       else
          printf("I lost: last bet %d\n", oplastbet);
   }
}

main()
{
   process bettor a, b;

   a = create bettor();
   b = create bettor();
   printf("May the bettor process win. Good luck!\n");
   a.init(b, 1, BETTING_LIMIT);
   b.init(a, 0, BETTING_LIMIT);
}
```

# OUTSIDE CONTACTS

**Electronic access to university community**

**(info@riacs)**

**Active cooperation with Stanford**

Joint Network Project with CASIS (multimedia "telescience")

Postdoctoral positions for CASIS graduates

Consulting professors teaching at Stanford

**Contacts with funding agencies**

DARPA, DoE

**Planning an industrial affiliates program**

# PROJECT "R"

RIACS Core Research Program is part of RIACS mission:

1. Pool of CS expertise to support NASA Ames.

2. Links to universities and industry.

3. Focused research effort on advanced computing environment to support scientific research in late 1980s and beyond.

# PROJECT "R" (con't)

## GOAL:

Very high level support of entire process of scientific investigation from problem formulation to results dissemination.

## COMPONENTS:

SCIENTIST'S AIDE: deals with user in language, concepts, pictures of discipline.

CONCURRENT PROCESSING SYSTEMS: virtual machines and software parts libraries for various models of parallel computation.

HARDWARE AND NETWORKS: actual machinery on which computations are performed.

# RIACS Research Accomplishments 1985:

1. Hypercube operational and serving as focal point for research and seminar in new algorithms for CFD (Chan) and CC (Partridge).

2. Initiated multiprocessor parallel machine testbed -- Sequent, Hypercube (Brown).

3. Developed experimental version of Concurrent C available for programming hypercube and other multiprocessors (Brown).

4. Developed prototype graphical program composition system for distributed computations spanning heterogeneous elements of networks (Brown, Denning).

5. Initiated first stage of concurrent processing research and parallel architecture simulation (Adams).

6. Formulated proposals, possibly joint with CASIS, on multimedia teleconferencing, privacy, and other issues in scientific networks (Leiner).

7. Initiated theoretical research in parallel algorithms and distributed systems initiated (Denning, Leiner, Raugh).

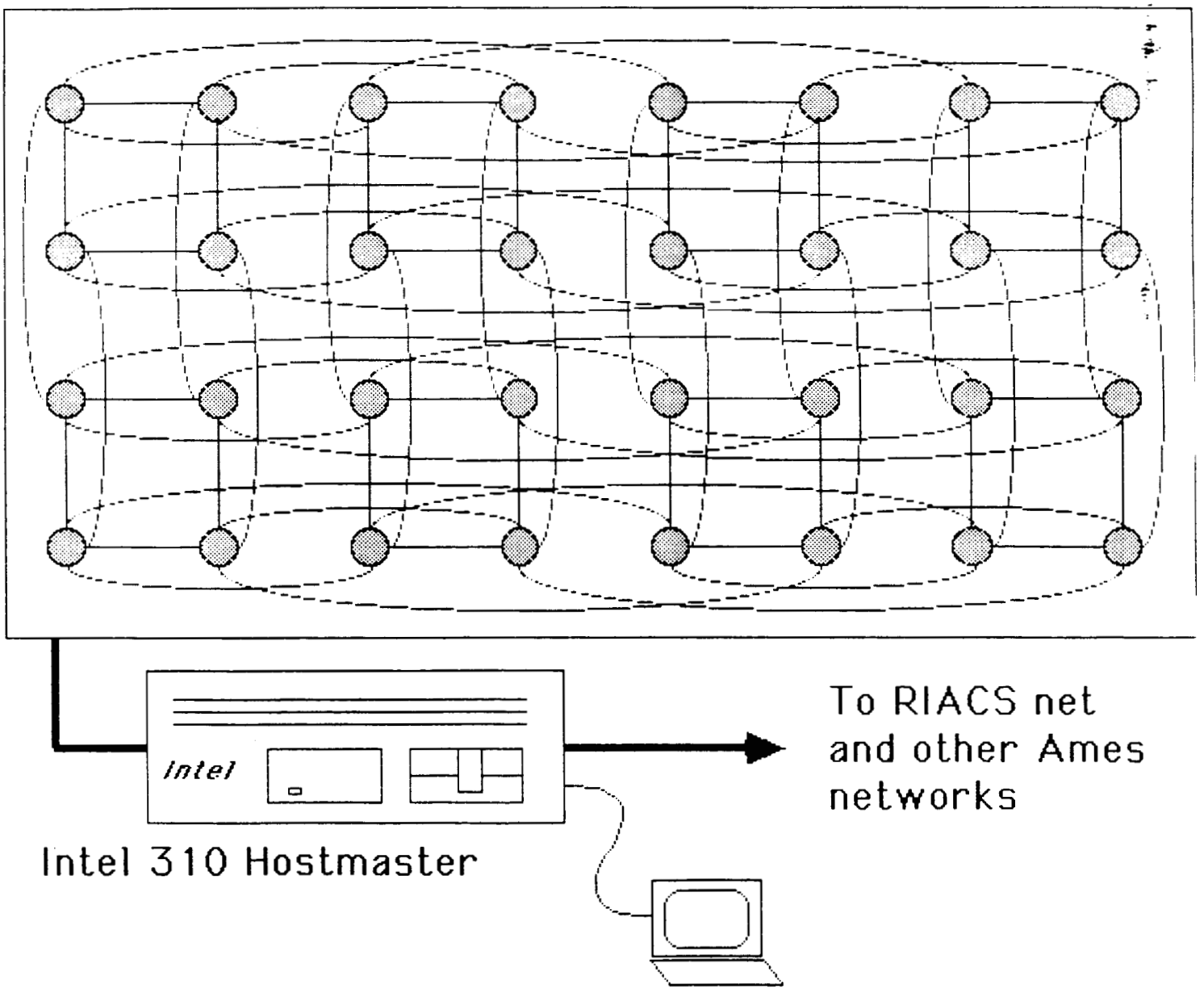
8. Initiated project on sparse distributed memory system. Simulators on Symbolics LISP machine and on hypercube will allow study of applications to robotics and autonomous systems. Breadboard model being designed in cooperation with Stanford (Kanerva, Raugh, {Flynn}).

The next six pages summarize an experiment by Bob Brown to assess the initial performance of the Intel hypercube and compare with the Sequent Balance 8000. The images are:

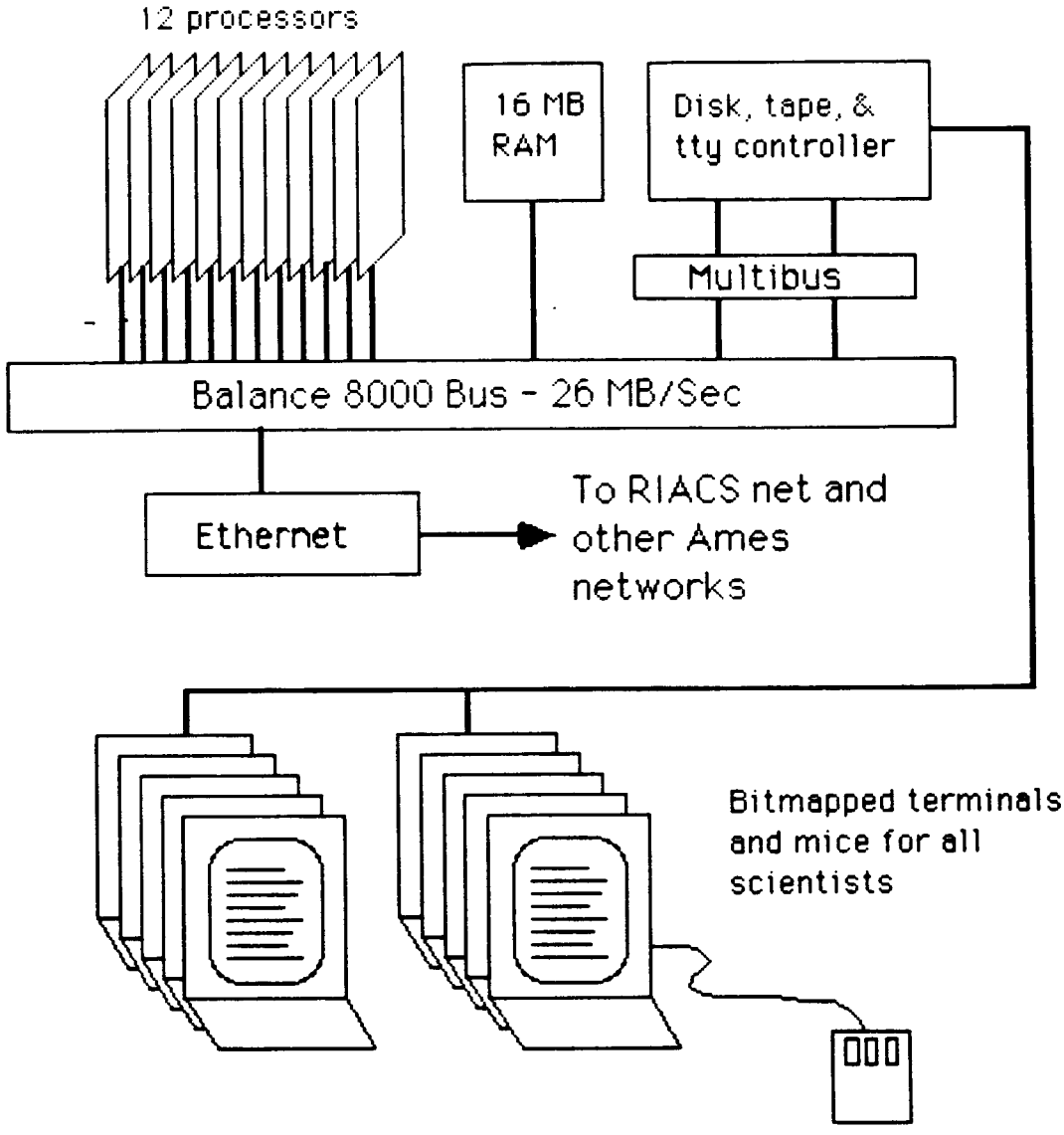
1. Schematic diagram of the RIACS 32-node hypercube. Each node has 512Kb of main memory and the machine has a total of about 16 Mb of main memory.
2. Schematic diagram of the RIACS Sequent computer. It has 12 processors sharing access to 16 Mb of main memory over a high-speed bus.
3. A partial image of the Mandelbrot set (see *Scientific American*, August 1985). This computation was chosen because it is very simple and requires no interprocessor communication. The 32 lines near the bottom of the figure are missing because the 32 processors on the hypercube computing them had not completed at the time the picture was taken.
4. A dataflow diagram of the computation of the Mandelbrot set for N processors. Each processor computes 1/N of the lines in the image and sends its results to a host process that displays them on a bitmapped screen. On the Sequent machine, a UNIX pipe funnels the N result streams to the terminal driver. On the Intel machine, the hypercube network funnels them to the Hostmaster processor, which relays them to the terminal driver. Obviously, there is a bottleneck in getting the output from the N processors to the one bitmapped terminal.
5. A graph of the real execution times to completion of the Mandelbrot set program for various numbers of processors. A single process version of the program took 100 minutes on the hypercube, 55 minutes on the Sequent, and 20 minutes on a VAX 780. A 10-process version took 5 minutes on the Sequent, which is the same as a 20-process version on the hypercube.
6. A graph of hypercube speedup ratios derived from the previous graph. The ideal of linear speedup is suggested by the straight dashed line. The actual curve is not straight because there was an ARPANET user logged in the hypercube during the overnight run to collect the data.

# RIACS Hypercube

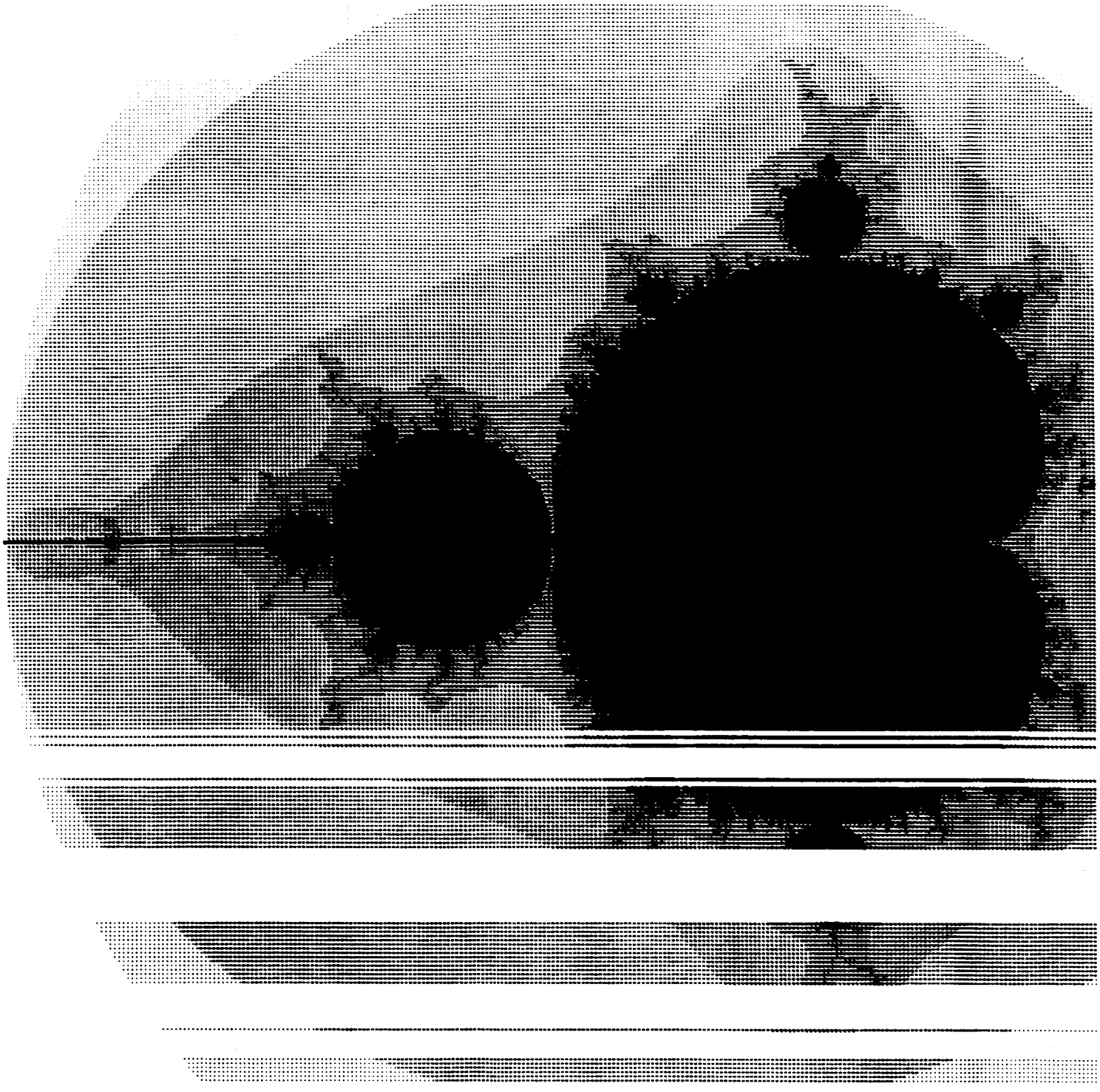
32 nodes, each is an Intel 80286/ 80287 chipset with 512 KB main memory.



# RIACS Central System

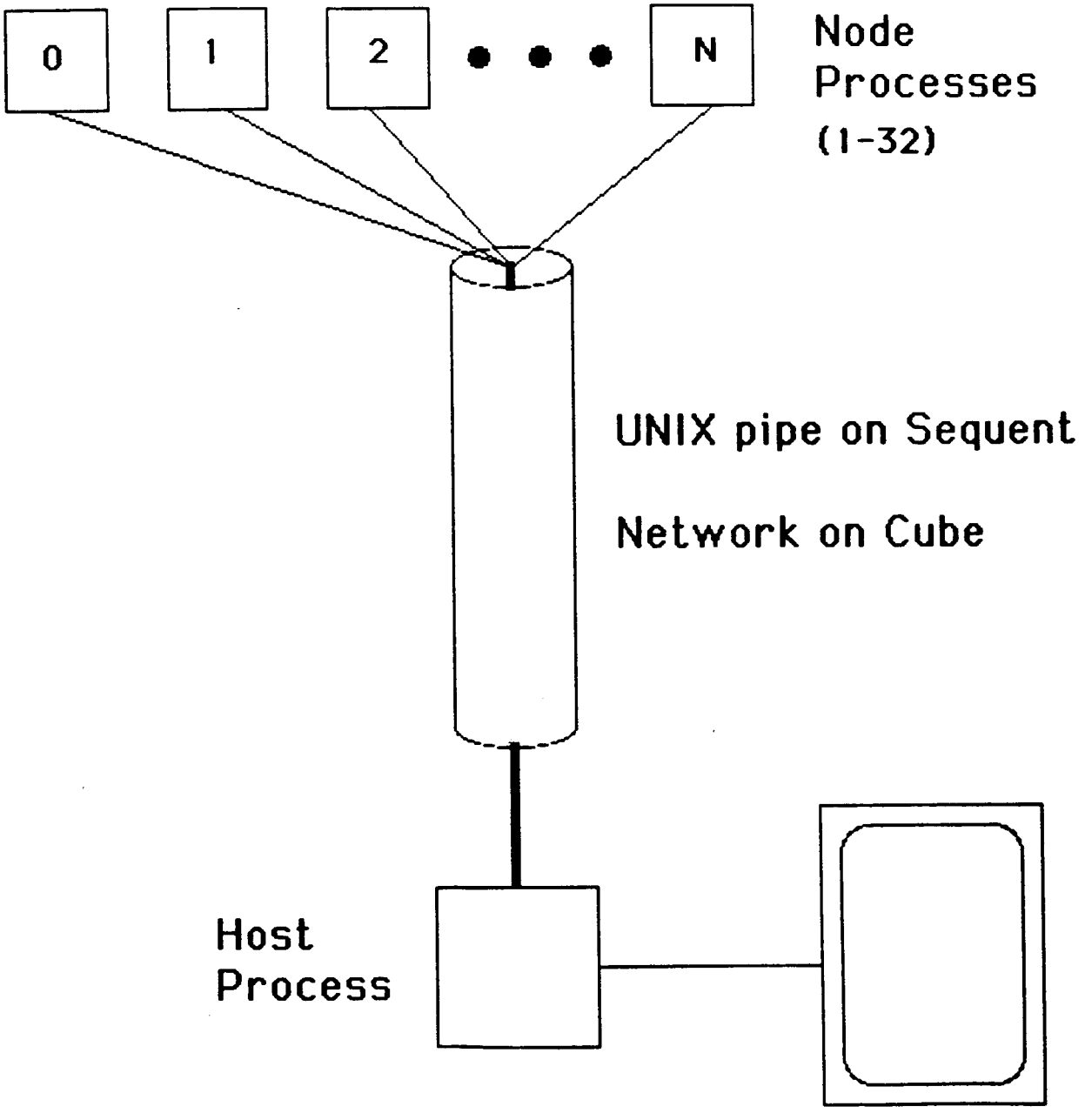


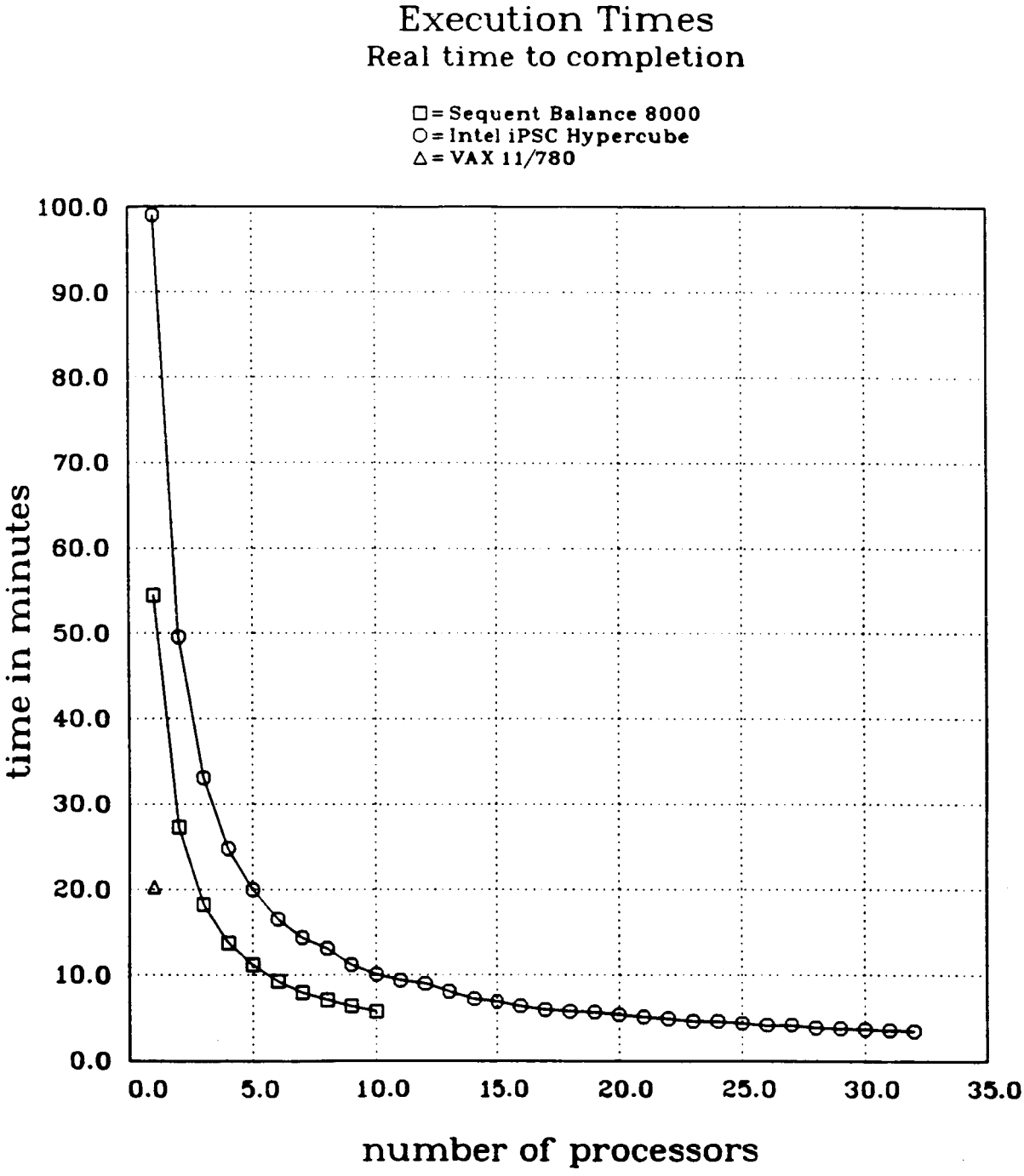

Tightly coupled multiprocessor supports "Concurrent C" project and multipurpose timesharing functions

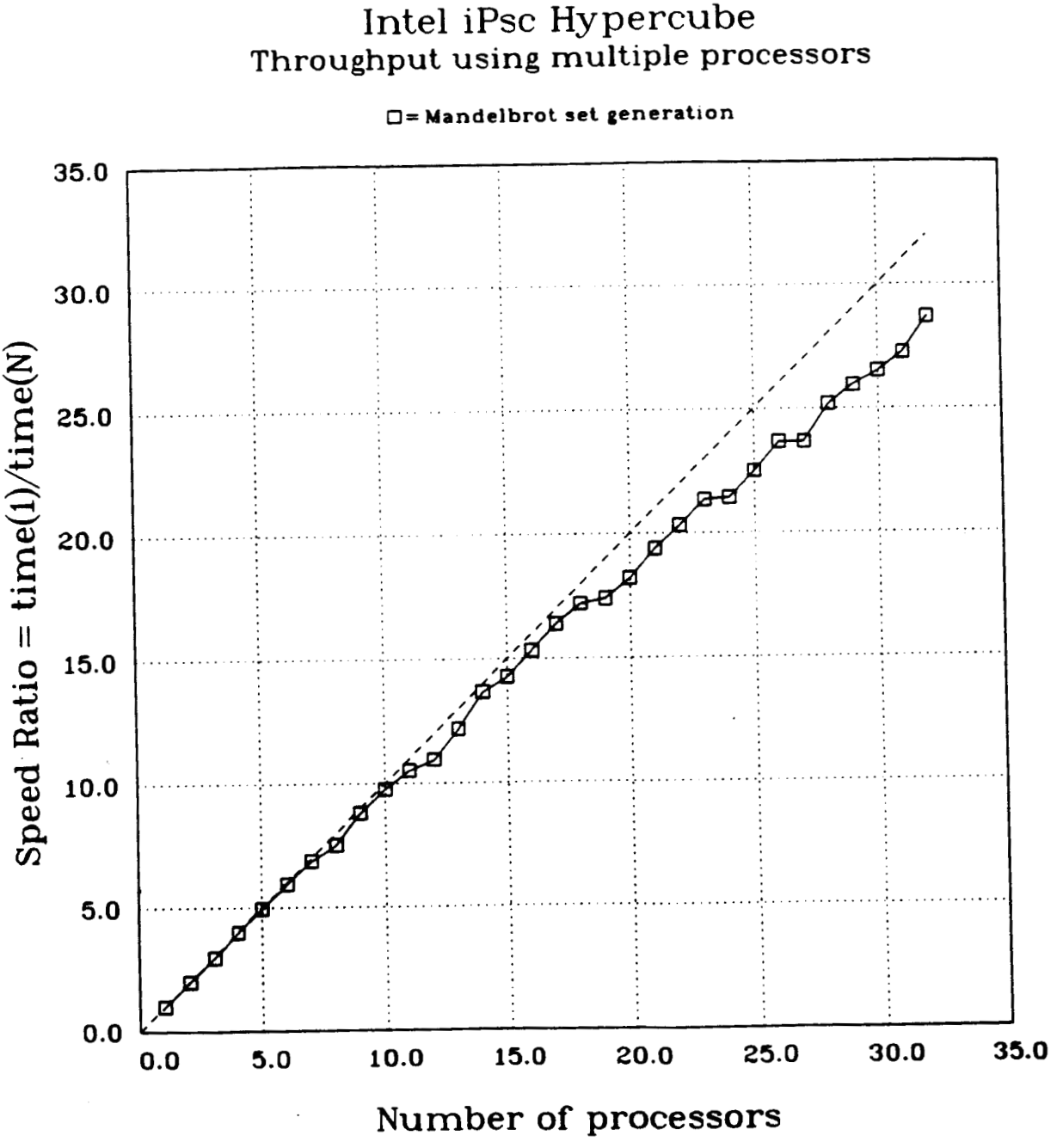


(hydra) print-screen

# Sample Concurrent Computation




Execution Times
Real time to completion
□ = Sequent Balance 8000
○ = Intel iPSC Hypercube
△ = VAX 11/780
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
time in minutes
0.0
5.0
10.0
15.0
20.0
25.0
30.0
35.0
number of processors


Intel iPsc Hypercube
Throughput using multiple processors
□= Mandelbrot set generation
Speed Ratio = time(1)/time(N)
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
35.0
Number of processors

# RIACS Near Term Goals:

1. Expand multiprocessor parallel machine testbed to continuously track state of the art in multiprocessors.

2. Complete development of programming languages and programming environments for large multiprocess computations.

3. Complete development of distributed program composition environment to demonstrate workstation–supercomputer computations, programs parts library, and network privacy.

4. Contribute to algorithms for multiprocessors in specific disciplines. Develop software tools for mapping from program spaces to specific machines.

5. Develop simulator for concurrent processing and parallel architectures.

...*more*

# Near Term Goals ... con't

1. Demonstrate basic telescience and teleconference facility over high-speed voice and data networks.

2. Contribute to advanced graphics techniques for visualizing results of fluid flow and chemistry computations, including nonlinear dynamics.

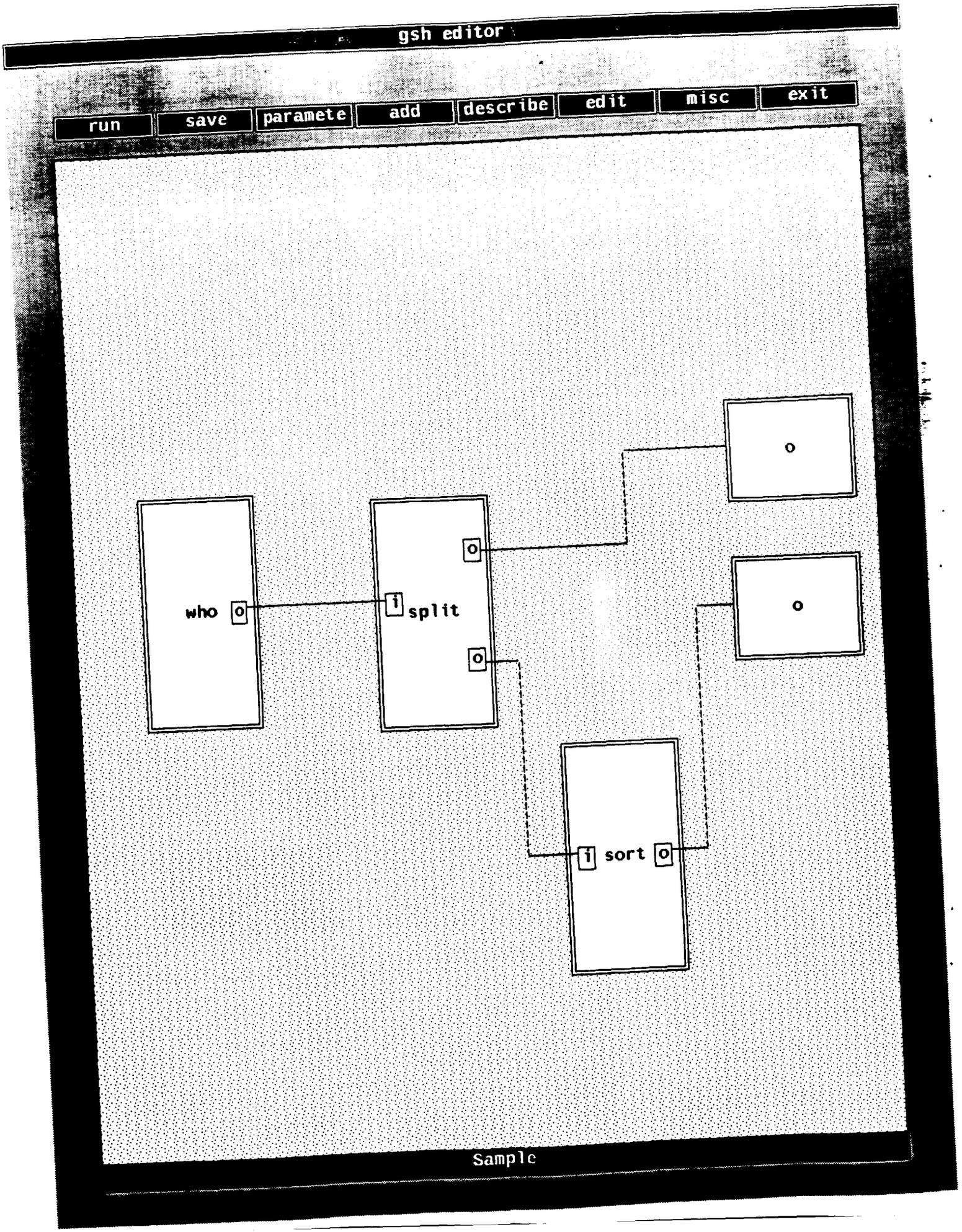
3. Demonstrate utility of sparse distributed memory system through working simulators with direct applications to robotics and human factors modeling. Breadboard model of memory devices.

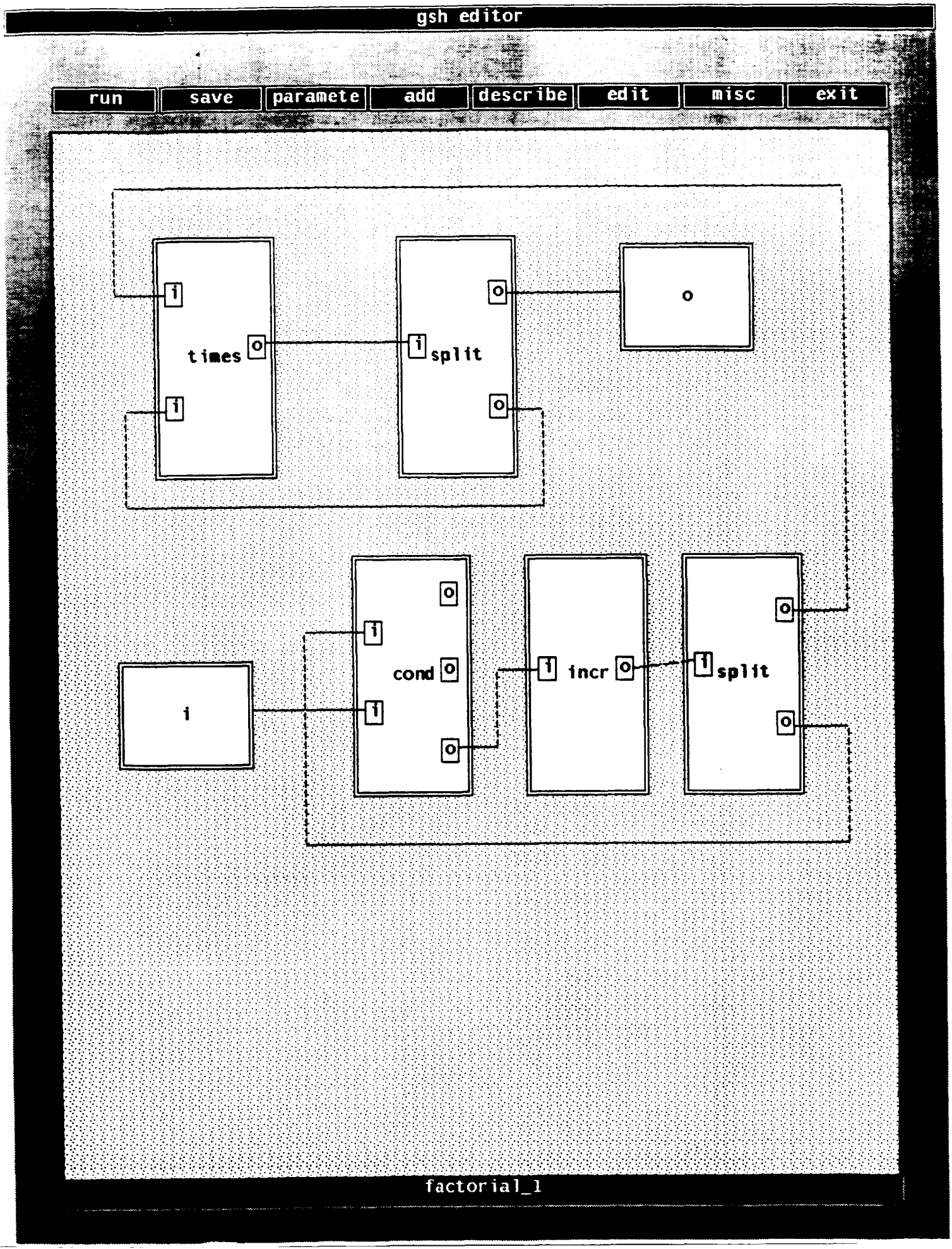
4. Assist in CFD expert systems.

The next two pictures are bitmapped screen images of programs run under the Graphical Shell (gsh). This is a prototype of a distributed program composition system that will operate with hetereogeneous networks. The prototype was developed during summer 1985 by Jeffrey Yost, a new graduate student at the University of Utah, with supervision by Bob Brown and Peter Denning. The first picture was drawn by positioning icons and connecting lines between them with the mouse. It represents a UNIX 'who' program, whose output is split into two steams, one of which is sent directly to an output and the other is sorted before being sent to the output. The network in this picture can be collapsed into an icon and used as a component of larger networks. The names at the top of the screen are for pull-down menus that aid in program composition and execution.

The second picture is a dataflow network that computes factorials. Both these programs actually run.

Future work includes the improvement of the interface, binding of icons to different machines on a network, and construction of a database for storing and locating program components.

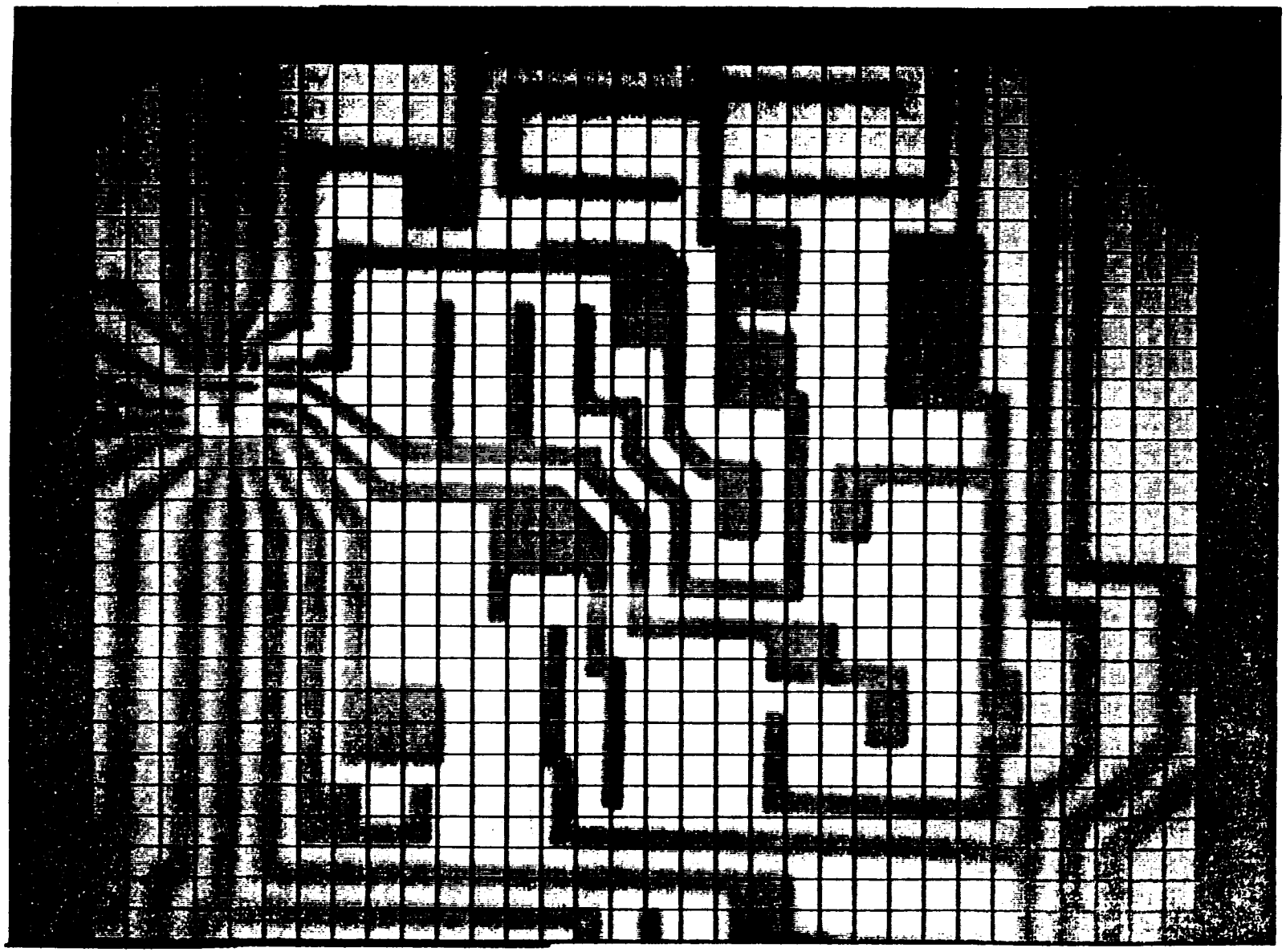

gsh editor
run
save
paramete
add
describe
edit
misc
exit
o
who
o
i
split
o
o
i
sort
o
Sample


gsh editor
run
save
paramete
add
describe
edit
misc
exit
i
times
o
i
i
split
o
o
o
o
i
cond
o
i
i
incr
o
i
split
o
o
factorial_1

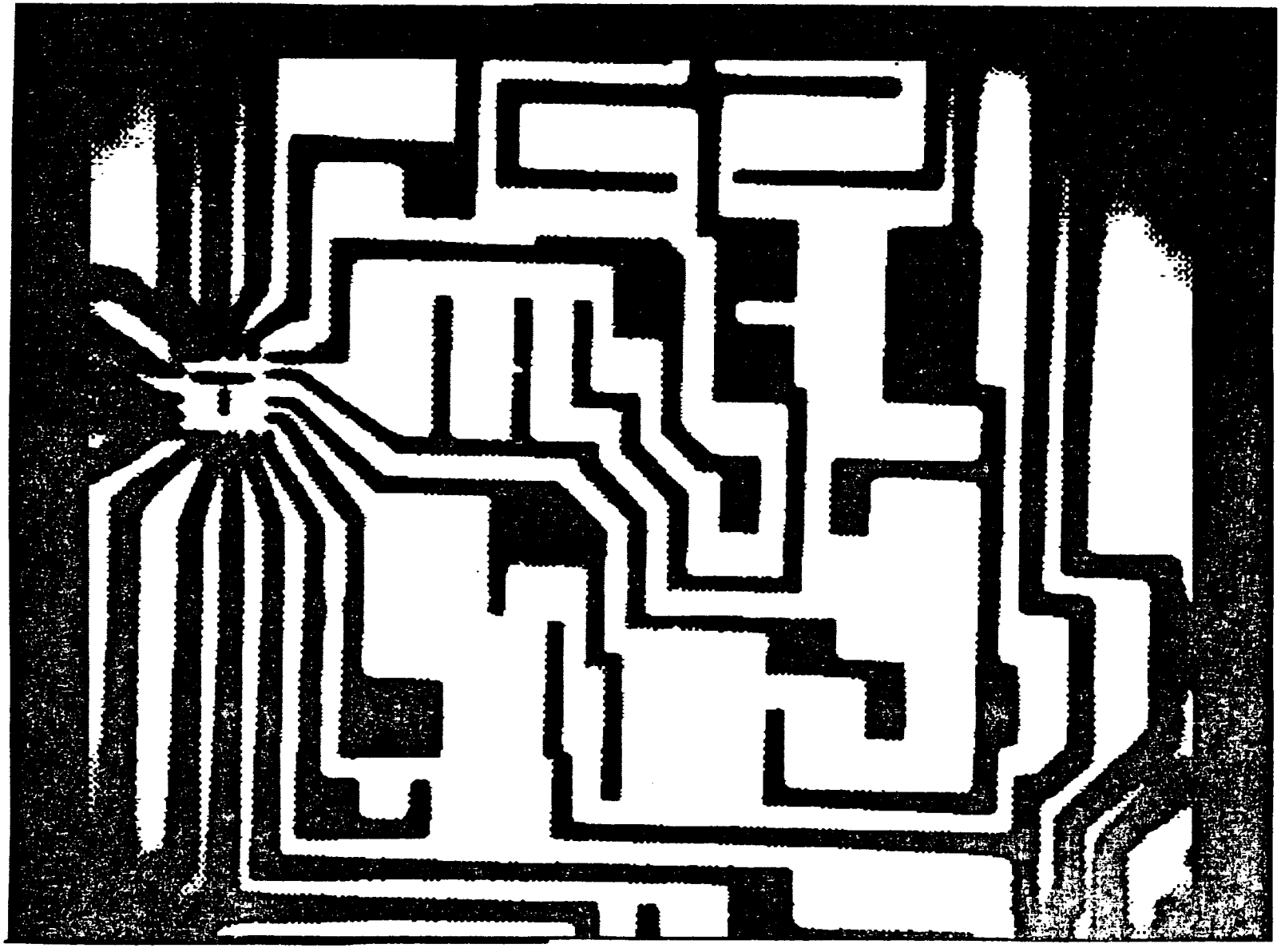
The next five pictures illustrate an image processing problem being studied by George Adams on the hypercube. The goal is to take an image of a printed circuit obtained under poor illumination – e.g., with a TV camera and flashlight looking into a space-station assembly -- then determine whether the circuit is broken. The images are:

1. A printed circuit under poor illumination.
2. An enhanced image of the board computed by using a single illumination threshold over the entire picture. This process gives poor results near the edges, which are in shadow.
3. An enhanced image of the board computed by segmenting the board into small squares and using a different threshold in each.
4. A contour of all the edges of the circuit constructed from the segmented image. If each segment is assigned to a different processor, there are interesting questions. How do neighbors match up their impressions of edges? How do edge-paths traversing many segments get labelled so they are recognizable as one path rather than many segments?
5. A summary of the problems new to the hypercube architecture from this problem.

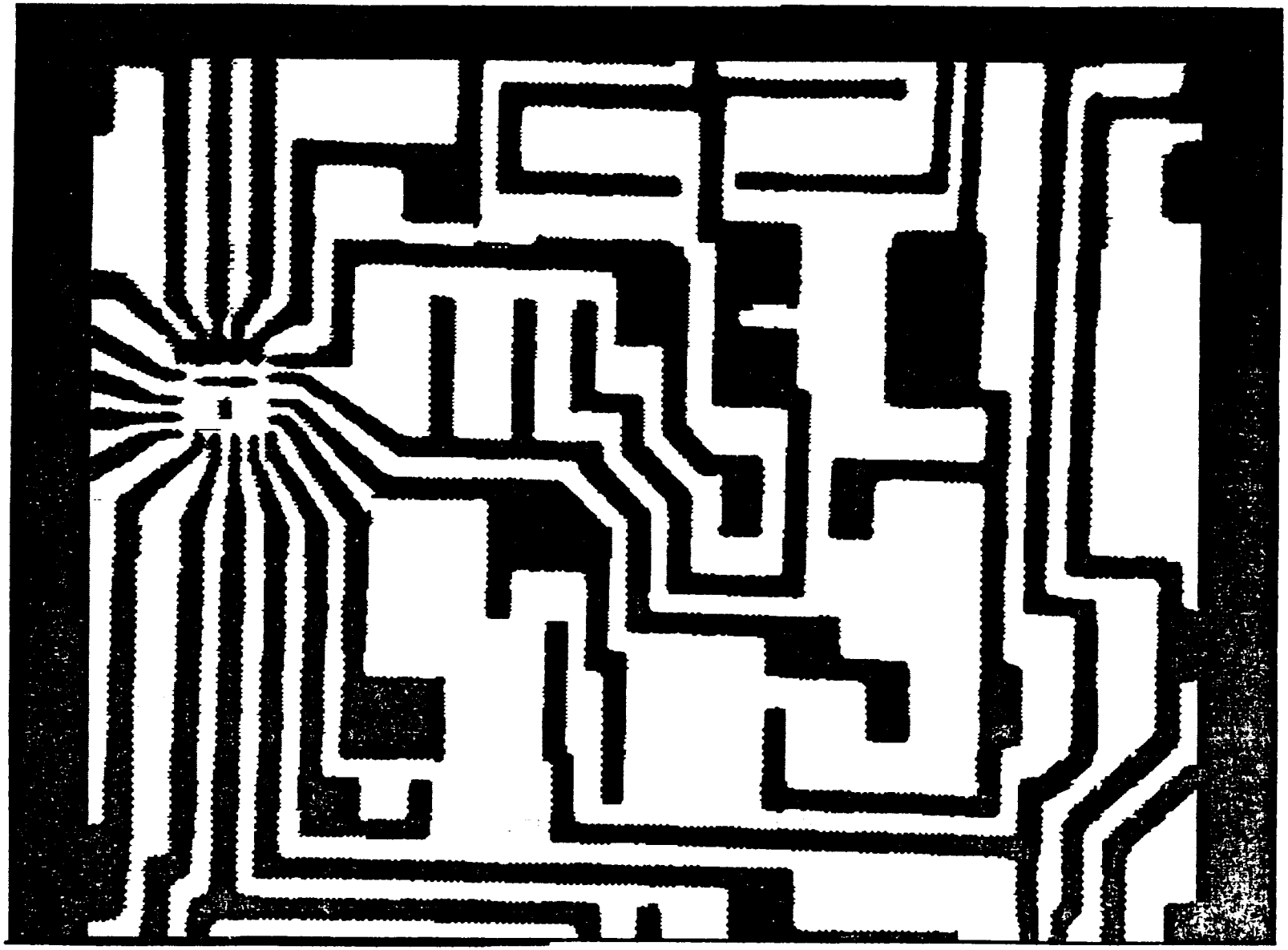
In addition to being an interesting image processing application in its own right, this project will be useful for studying mappings of parallel algorithms onto parallel machines. It represents a worst case: the load on each processor and the communication patterns are data dependent.



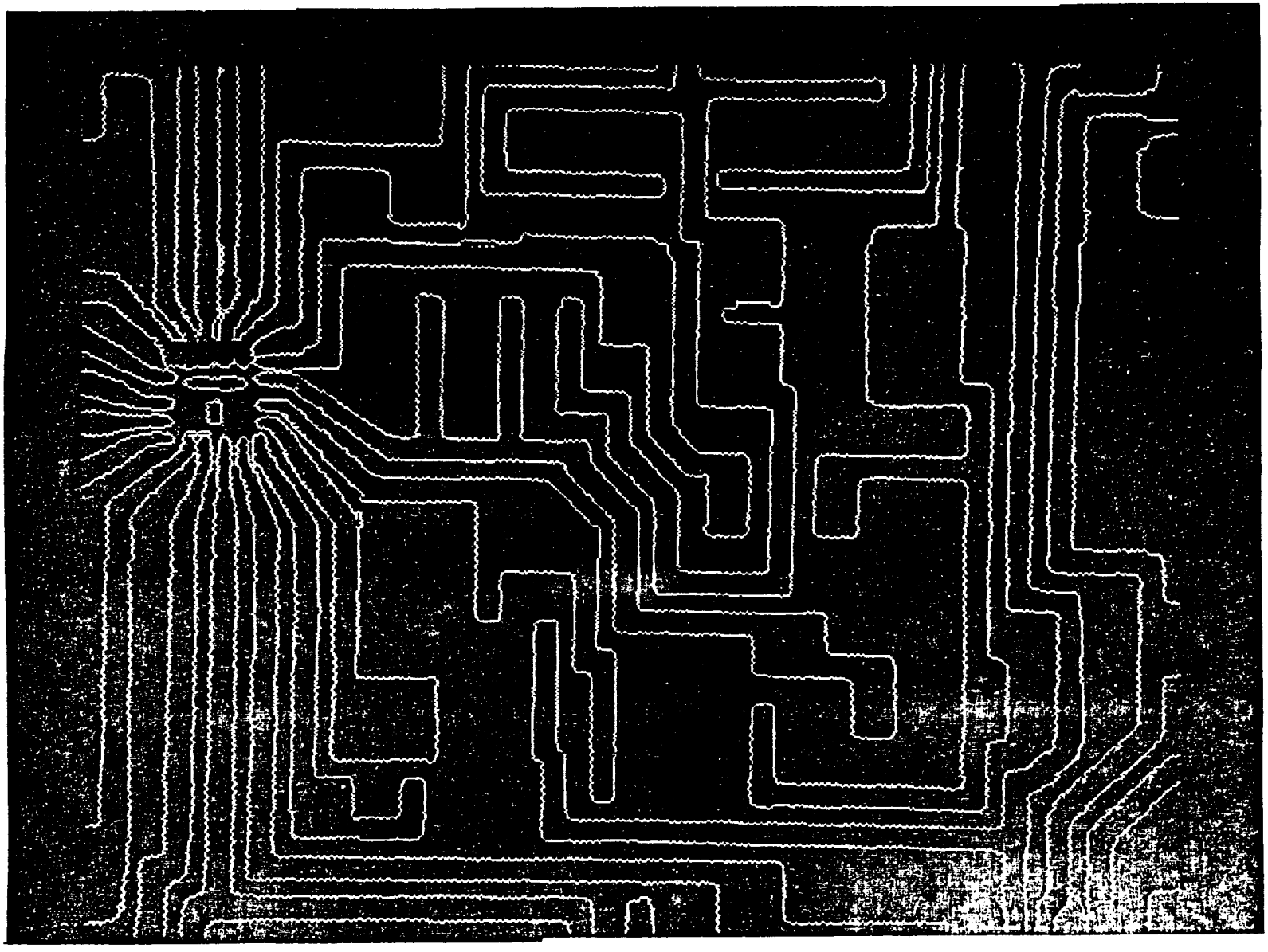
Section of an intentionally poorly illuminated circuit board overlayed with a 16x16 pixel grid.



A binary image obtained by thresholding the original image with a single threshold of 153.



Binary image resulting from segmentation of the original image using edge-guided thresholding



Contours extracted from the segmented image.

# IMAGE CONTOUR EXTRACTION

New to Hypercube

NOT a collection of loosely coupled parallel processes

Execution time, communication load, and communication pattern all strongly data dependent

Different images will stress Hypercube in different ways, e.g., load balancing, communications